Status Report

Compression and Progressive Transmission of Digital Images

NAG 5-2694

for the period through December 1995

Principal Investigator
Jeffrey W. Percival

University of Wisconsin-Madison
Space Astronomy Laboratory
1150 University Avenue
Madison, WI 53706

Status of Research under NASA Grant NAG 5-2694 through December 1995

We summarize the first year of work on this three-year award.

Our proposed milestones were rearranged at the beginning of work from the traditional sequential development model to the more modern and effective rapid prototyping model. In this revised approach, we proposed to develop a complete end-to-end prototype of the Progressive Image Transmission (PIT) system in the first year, and to release it to selected users for beta testing. This revised approach would allow important user feedback before undertaking the big task of designing the ultimate user interface.

We succeeded in developing this prototype and have distributed it to a select group of scientists involved in both professional and educational work. We have received the first round of feedback, which is already influencing our next task, that of redesigning the compressed image file and transmission protocol. This redesign, to be completed by the middle of 1996, will allow digital image libraries to store images more effectively and to serve them out to PIT users with less overhead.

The current PIT system was developed using the Tk graphical scripting language. We chose this because of its simplicity and flexibility. We are now considering, and will probably choose, the web-based JAVA language as the ultimate implementation tool. This will, in one fell swoop, give us machine independence and a Web-oriented browsing mode that will guarantee the role of our Progressive Image Transmission in the emerging Web universe.

We are pursuing Progressive Image Transmission interests in the educational realm (e.g. UNM's LODESTAR program), the scientific realm (e.g. NCSA, WIYN Telescope, CTIO Observatory), and the private sector (3M Electronic Imaging).

We have been making detailed quarterly progress reports to our Technical Officer, and append them here as a detailed description of work and progress.

— Quarterly report for the quarter Oct-Dec 1995 —

October marked the beginning of our effort on this project. We learned that we had been funded at 80the best way to proceed was to start out a little more slowly than we had planned. We deferred the hiring of the programmer, pending our reevaluation of the schedule and milestones. We anticipate being able to meet all of our milestones and produce all of our deliverables, albeit with some modifications to the detailed sequence of events.

We began our research into Graphical User Interface (GUI) builders. A literature search yielded a number of good candidates, all within the amount budgeted within the proposal. One question we will have to answer is whether we want a simple GUI builder or a more sophisticated cross-platform development kit. The simple GUI builders have the advantage of simply accelerating the construction of a Motif-based GUI, while the cross-platform kits achieve portability by *emulating* the target's environment, rather than producing code *for* it. This means, for example, that the programmer's GUI specification exists in what may be a vendor-specific, proprietary format rather than, say, a simple UIL file that can be ingested by some other GUI builder. In other words, the effort that goes into using a cross-platform kit may not be portable to kits from other vendors. They lock you into their product. We are tending to think that requiring UIL as an output product is the way to go, as we would end up with a portable GUI specification, requiring only that a target support the traditional Motif apparatus and a C compiler. We remain open on this issue, and are continuing our investigation.

We were asked to prepare and present a colloquium and demo for the Project at GSFC. Co-investigator Richard White described the compression technique underlying our Progressive Image Transmission (PIT) scheme, and the PI presented results from the prototype implementation. We were asked to quantify the effectiveness of the compression for non-astronomical images, specifically weather images. We agreed to sample some images provided by the Project and report on the compression performance. In the demo the PI showed the prototype PIT system, transferring images from Wisconsin. Dr. White used IDL to demonstrate in greater detail the nature of the lossiness, and quality of the reconstruction, of compressed images.

The PI met with the RSPAC PI and briefly discussed the RSPAC block diagram and how we could best coordinate our efforts with RSPAC.

— Quarterly report for the quarter Jan-Mar 1995 —

In January the Project help a PI workshop to allow the various CAN efforts to learn about each other. The PI could not attend because of the impending, and ultimately successful, birth of our beautiful second daughter, Ruth. Our project was represented by the Co-I Dr. White, who presented the results of his investigation of his compression performance on non-astronomical images. Specifically, he looked at a GOES image converted from GIF into FITS format, and a picture of President Clinton. He reported that these images performed as well as astronomical images under aggressive levels of compression, and that, as expected, the lossiness was nearly undetectable with only 10data preserved. The GOES weather image clearly showed the cloud cover, which seemed to be of particular interest at the colloquium in December.

The PI prepared a Web page describing our project. The URL is:
http://jerry.sal.wisc.edu/ jwp/can.html

This page presents the abstract and technical sections of our proposal, further references for both the wavelet compression and the Progressive Image Transmission (PIT) prototype system, and the examples presented at the GSFC colloquium in December and the CAN workshop in January. Also included are links to the home pages of the PI and Dr. White, as well as a link to the Digital Library Technology (DLT) Project.

We were asked to deliver our prototype (non-GUI) system to the Project (contact: Upendra Shardanand, shard@ds.gsfc.nasa.gov) for experiments, playful use, demos, etc. We had numerous exchanges with Mr. Shardanand, and think that he is now up to speed on the use of this prototype.

The PI and Dr. Lal, the DLT Project Manager, discussed a reorganization of milestones. Instead of working sequentially from the PIT protocol through to the client and server, culminating at the end with a Graphical User Interface (GUI), we decided instead to implement a rapid-prototyping cycle. We will generate a simple GUI quickly using the Tk graphical scripting language, allowing both rapid feedback into the protocol specification and early use of the PIT system by the Project and RSPAC. We will generate a series of "mini releases", with each released GUI having an incrementally larger functionality. We hasten to emphasize the following points:

1. The mini-releases must be understood to be simple, quickly produced "rough drafts" of the end product. 2. The mini-releases to not necessarily represent the final deliverables in appearance, capability, performance, or even language of implementation. 3. Their use within and outside the project must strictly controlled unless points (1) and (2) above are clearly

understood.

We simply fear that if presented out of context, these crude approximations to our goal will be misconstrued as representative of the final capability or quality of our deliverables, and wish to avoid any such misunderstandings. This reordering is already bearing fruit, and we hope to release our first rough-draft GUI within a week or two. We will contact Dr. Lal when this mini-release is ready.

— Quarterly report for the quarter Apr-Jun 1995 —

This quarter saw the completion of the end-to-end prototype, including a simple Xt-based image viewer. This simple, portable viewer receives image data via shared memory, thereby eliminating the previous inefficiency of writing the image to disk and re-reading it into a separate viewing program.

Serious effort was devoted to portability issues. The code was upgraded with ANSI subroutine prototypes, and BSD-unix dependencies were eliminated or worked around via conditional compilation. With RSPAC computer support, I created a working Solaris version. Because Solaris is a standard System V product, I expect that porting to other System V systems such as HPUX should proceed smoothly.

Currently supported systems: DEC OSF/1 (alpha) DEC Ultrix (mips) DEC Ultrix (vax) SGI IRIX Sun Solaris Sun SunOS

Near term goals: 1. delivery to RSPAC (done as of 25 July 1995) 2. delivery to NASA DLT (Robert Lehr) 3. add metrics to transmission (how much, how soon, how good...) 4. Attack code bottlenecks for improved efficiency 5. Solicit feedback from RSPAC and DLT users 6. Assess image viewer, write requirements for next version. 7. Consult with possible collaborators at NCSA 8. Consult with possible collaborators at Harvard's Center for Astrophysics (already involved in CAN).

— Quarterly report for the quarter Jul-Sep 1995 —

This quarter saw the beta release of the Progressive Image Transmission system. Initial feed back was good, with a number of suggestions for immediate improvement. These include display of transmission metrics (in progress since last quarter), improved file choosing (file size, title, etc.) and image subsetting (sending a piece of the image that contains an area of interest).

Typical activities this quarter were compiling for multiple platforms, walking through beta sessions with users, promoting contacts with developers in related areas (more below), and various coding efforts such as switching to the System V signals interface and experimenting with protocol metrics.

Outside contacts: I am still pursuing outside contacts. I sent reprints to the NCSA folks, and will recontact them this month. The 3M company held a "Technology Transfer Fair" in St. Paul, at which our project was represented by UW Industrial Relations people, and we ended up with four 3M groups interested in further information. These groups cover the Electronic Imaging, Advanced Media and Systems, and Printing and Publishing Divisions.

Code development: I spent some time experimenting with transmission metrics, including a robust baud rate estimation. This is important, because it influences the "time to go" metric and because small errors in estimating the instantaneous transmission rate become highly magnified in "time to go" when the user is in the highly compressed regime in which our system excels. The problem is estimating a rate in the presence of noise (timing jitter), and I experimented with an algorithm I ran across in my telescope control systems work. This algorithm is called the "alpha-beta radar tracker", and is a discrete-time Z-transform method of estimating the speed of a radar target in the presence of noisy data. The algorithm sports two independent variables controlling gain and damping factor. The gain controls how quickly baud rate changes are sensed, and the damping factor controls noise (timing jitter) susceptibility. These tests gave promising results, and appear to be better than simple box-car or sliding window averages, but I will continue to devote a small fraction of time to this in the next few weeks.

Another interesting small-scale effort was an experiment in video transmission. Our Progressive Image Transmission system so far has focussed on single digital images, but the extension to video sequences (e.g. television) is simple. A grabbed frame becomes a digital image, and our GUI only needs a few extra control buttons (pause, skip, and so on) to become what

I call a "video frame browser". This is *not* a videoconferencing tool, or a real-time video tool, rather a simple scheme for sending individual video frames in which the user can control resolution and image quality on a per-frame basis. This simple variation on our baseline GUI could be used, for example, by students to capture sample frames from a real-time video from an observatory, laboratory, or other interesting video source. I point out that the protocol, server, and client interface all use the same protocols and algorithms as the digital image system, and still target the same ultra-low bandwidth group of users, providing a very resource-efficient spin-off of the basic program.

Main upcoming issues:

1. new GUI client. We budgeted a new hire for this, a new workstation, and a commercial GUI builder program. We need to launch this activity, but I would like to make sure that FY96 funding is stable and in place before committing to this.

2. Is Mosaic the way to go any longer? What about Java, which as I understand it externalizes algorithms for a WEB program rather than internalizes them? Perhaps this sort of thing could be discussed at the upcoming PI meeting?

Near term goals: 1. new release of beta system 2. solicit feedback from RSPAC, DLT, and other users 3. specify and start new GUI client 4. consult with NCSA 5. consult with Harvard's CFA (already involved in CAN) 6. learn about Java 7. identify code bottlenecks for improved efficiency

—— Quarterly report for the quarter Oct-Dec 1995 ——

This quarter saw several updates to the beta release, and we pursued several new contacts interested in progressive transmission. We ran a demo with the LODESTAR program (University of New Mexico) in which students run robotic telescopes from schools. The LODESTAR program features video signals from observatories sent over high-bandwidth lines, but of course such connections will not exist for many rural schools. We promoted our "Progressive Video Transmission" system (see last quarter's report) as a useful tool for isolated classrooms. In addition, the Cerro Tololo Interamerican Observatory (CTIO) in Chile has expressed interest in using the Progressive Image Transmission (PIT) system for scientists using the CTIO telescopes over slow transcontinental telephone lines. We will install the PIT system there this week, and keep in active contact with them as their users begin to use the system.

Code developments:

We implemented the transmission metrics and progress indicators, allowing users to see how things are going, how much time until a new view is ready, how much time to complete, etc. These features met with universal acclaim.

The biggest single unimplemented feature coming out of our beta testing is "regions of interest" (ROIs), in which a user can draw a box around a part of an incoming image and get a higher priority assigned to that area. This feature is a natural one for PIT, as the very rapid appearance of the initial image lets the user identify interesting ROIs quickly.

This affects the code development in three ways:

1. Easily identifying the ROI (a GUI issue) 2. Easily extracting the ROI (a server issue) 3. Sending ROI data intermixed with regular data (a protocol issue)

Item (1) is financially interesting because GUI development is a big thing, and we have deferred starting this due to budget uncertainties. We need to make this hire, and for that I'd like to have the budget uncertainty removed (as would everybody, of course).

Item (2) is technically interesting because the ROI must be extractable whether or not the raw image is available. i.e. if an archive has been stored as wavelet files, we do not want the server to have to invert the transform to get the ROI. We have a handle on how to do this.

Item (3) is not anticipated to be a big deal, but items (2) and (3) taken together have caused us to consider a quantum leap in server code: up until now, the wavelet file format has been very different from the protocol

format. We now intend to merge these two things, and design a progressive file format. This will be neat, reducing the server overhead and making file truncation equivalent to progressive transmission. The PI will meet in Baltimore with the Co-I Rick White in the next few weeks for a several-day design session.

GUI directions:

Everyone is now convinced that JAVA must be explored, and rumored JAVA speed improvements may remove any down side to a JAVA-based PIT. I have asked a systems programmer here at the Space Astronomy Lab to begin to develop a simple GUI, which we can later enhance with a full-blown PIT system when the budget problem goes away.

Budget:

We have been underspending due to the delay in GUI work. We hope that the apparent surplus does not give the impression that the grant can be reduced further, because we will have to work double-hard in the future to make up for the delays this fall and winter. Also, moving to a Web-based JAVA applet, while clearly the thing to do, involves more unknowns than the traditional GUI we have been planning thus far. We hope to get a handle on the level of effort from our JAVA startup effort mentioned above.

Reports:

We are late with our annual performance report to the Grants Officer, and will submit one by 12 January 1996.